A Nonlinear Model Predictive Control Algorithm for Obstacle Avoidance in Autonomous Ground Vehicles within Unknown Environments

Jiechao Liu, Paramsothy Jayakumar, Jeffrey L. Stein, and Tulga Ersal

***Abstract*—A nonlinear model predictive control algorithm is developed for obstacle avoidance in high-speed, large-size autonomous ground vehicles (AGVs) that perceive the environment only through information provided by on-board sensors. The mission of the AGV is to move from its initial configuration to the goal configuration safely. The resulting trajectory should be collision-free and the AGV should be dynamically safe. As a starting point, the scenario where the vehicle moves on a flat surface at a constant speed is considered. The nonlinear MPC algorithm generates steering commands for completing the mission while enforcing safety constraints. The first safety constraint is avoiding obstacles. This is fulfilled by constraining the position of the AGV inside a safe region established from sensor data. The second safety constraint is ensuring dynamical safety. This is translated into avoiding single tire lift-off, which is implemented by limiting the steering angle within a range obtained using a 14 DoF vehicle dynamics model. At each sampling time, at least one multi-phase optimal control problem (OCP) is formulated and solved on-line. The safe region is partitioned into multiple sub-regions, which can then be specified without using piecewise functions. The fact that the optimal trajectory traverses the sub-regions sequentially and hence the position constraints are different from phase to phase makes the OCP multi-phase. The multi-phase OCP is transcribed into a nonlinear programming problem using the *hp*-pseudospectral method, and solved using the interior-point method. Simulations of an AGV approaching multiple obstacles show the effectiveness of the proposed algorithm.**

***Index Terms*—Collision avoidance, vehicle dynamics, model predictive control, autonomous ground vehicles.**

I. Introduction

AUTONOMOUS ground vehicles (AGVs) are gaining importance and finding increased utility in both military and commercial applications. Although earlier AGV platforms were typically exclusively small ground robots, recent efforts have targeted passenger vehicles and larger size platforms, as well. For this size of vehicles, it becomes especially important to take the dynamical limitations of the vehicle into account to guarantee its dynamical safety during obstacle avoidance maneuvers [1]. Therefore, obstacle avoidance algorithms are needed that can ensure vehicle safety even if the vehicle is operating at its dynamic limits.

Many obstacle avoidance algorithms have been developed in the literature that allow for fast, continuous, and smooth motion of AGVs among unexpected obstacles. They can be classified into four categories: graph-search based methods [2], [3], virtual potential field and navigation function based methods [4], [5], meta-heuristic based methods [6], and mathematical optimization based methods [7], [8]. Among these categories, mathematical optimization based methods are particularly attractive, because they offer a rigorous and systematic approach to take vehicle dynamics and safety constraints into account.

A mathematical optimization approach can be used either in open-loop, if the environment is fully known *a priori*, or in closed-loop with a feedback controller for a more robust solution. Regarding the latter, the model predictive control (MPC) approach is one of the most widely adopted techniques [9]. Prior research has demonstrated successful applications of MPC to obstacle avoidance in AGVs [10], [11], [12], [13], [14], [15]. Some active safety methods leverage the MPC framework, as well, to ensure, for example, safe lane keeping or vehicle stability [16], [17], [18], [1].

The first applications of MPC to obstacle avoidance in AGVs assumed that the controller has full knowledge about the environment. They also were not concerned with the level of fidelity that the model used by the controller needs to possess for satisfactory performance, where the performance criteria in some cases also include the dynamical safety of the vehicle, such as no tire lift-off. Previous work by the authors aimed to address this gap by developing an MPC formulation that takes into account the information about the environment as provided by the on-board Light Detection and Ranging (LIDAR) sensor [19]. They also investigated the role of model fidelity and showed that the vehicle's dynamical safety can be guaranteed by limiting the steering angle using a high fidelity model, which then allows the MPC to work with a low fidelity model for trajectory optimization [19]. However, this investigation was done using an exhaustive search approach on a coarse mesh of the control input.

In this paper, we extend the work presented in [19] and [20], and develop a novel nonlinear MPC algorithm for obstacle avoidance of AGVs that can achieve an optimal and smooth operation of the vehicle through the obstacle field

Manuscript submitted February, 2015. The authors wish to acknowledge the financial support of the Automotive Research Center (ARC) in accordance with Cooperative Agreement W56HZV-14-2-0001 U.S. Army Tank Automotive Research, Development and Engineering Center (TARDEC) Warren, MI.

J. Liu, J. L. Stein, and T. Ersal are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: ljch@umich.edu; stein@umich.edu; tersal@umich.edu).

P. Jayakumar is with the U.S. Army RDECOM-TARDEC, Warren, MI 48397 (email: paramsothy.jayakumar.civ@mail.mil)

Disclaimer: The views expressed in this article are those of the author and do not reflect the official policy or position of the Department of the Army, DOD, or the U.S. Government.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
24 APR 2015		**00-00-2015 to 00-00-2015**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Nonlinear Model Predictive Control Algorithm for Obstacle Avoidance in Autonomous Ground Vehicles within Unknown Environments	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **US Army RDECOM-TARDEC,6501 E. 11 Mile Road,Warren,MI,48397-5000**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
See Report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**14**	

while ensuring vehicle safety. Optimality here refers to the optimality of the solution within the prediction horizon, taking into account all the information available at that moment in time, and not to the optimal solution that would have resulted if all environmental information was available for all times. Optimality is achieved by formulating the obstacle avoidance problem into an optimal control problem (OCP), which is then converted into a nonlinear programming (NLP) problem using the *hp*-pseudospectral method, and solved using the interior-point method. Two types of safety constraints are included in the formulation. The first type of safety constraint is avoiding obstacles. This is fulfilled by constraining the position of the AGV inside a safe region established from sensor data. The safe region is partitioned into multiple sub-regions, which can be specified without using piecewise functions. These specifications can then be used in the OCP formulation. The second type of safety constraint is ensuring dynamical safety. This is translated into avoiding single tire lift-off, which is implemented by limiting the steering angle within a range obtained using a 14 DoF vehicle dynamics model. Simulations of an AGV approaching multiple obstacles show the effectiveness of the proposed algorithm.

Throughout the paper, the following assumptions are made:

(a) All obstacles of interest are at least the height of where the LIDAR is mounted on the vehicle, which is in front of the vehicle.
(b) The vehicle longitudinal speed is maintained to be constant.
(c) The vehicle travels on a constant-friction flat surface.

Because the formulated OCP is nonconvex, it is not guaranteed that the solution from the OCP solver is the unique global optimal solution over the prediction horizon. Thus, the terms "optimal trajectory", "optimal states", and "optimal control" in this paper refer to the local optimal solution generated by the OCP solver, which is the first minimum it finds.

The rest of the paper is organized as follows. Section II introduces the basic principle of MPC briefly. Section III presents the formulation of the multi-phase optimal control problem for obstacle avoidance, including safe region partition approach, details of the vehicle dynamics model used, maximum steering angle establishment, and solution techniques. Section IV presents and discusses the simulation results. Conclusions are drawn in Section V.

II. Basic Principle of MPC

The idea of model predictive control is to utilize a model of the system to be controlled to predict and optimize future system behavior. MPC is an optimal control based state-feedback controller. The feedback law is obtained by an iterative on-line optimization over a moving finite prediction horizon. One advantage of using a moving time horizon is the ability to perform real-time optimization with hard constraints on plant variables [9].

The basic principle of MPC is illustrated in Fig. 1. At time t_0, starting from the state measurements, an optimal control sequence $\boldsymbol{\zeta}^*(t), t \in [t_0, t_0 + T_p]$ is computed by solving an

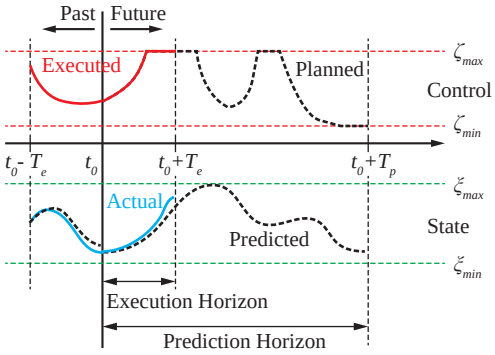


Fig. 1. Basic principle of MPC

open-loop, constrained, finite-time optimal control problem over the prediction horizon T_p. The control calculations are based on both future predictions and current measurements. The optimal control sequence $\boldsymbol{\zeta}^*(t)$ is bounded by the input saturations, and the resulting estimated optimal states $\boldsymbol{\xi}^*(t)$ satisfy the pre-defined constraints and minimize the cost function. Although the optimal control sequence is calculated over the horizon $t \in [t_0, t_0 + T_p]$, only a portion of the computed control sequence $\boldsymbol{\zeta}^*(t), t \in [t_0, t_0 + T_e]$ is sent to the plant and executed, where T_e is called the execution horizon and represents the portion of the computed optimal sequence that is implemented. Due to model simplifications, model parameter uncertainties, and / or other types of noises and uncertainties, the actual states of the system $\widetilde{\boldsymbol{\xi}}^*(t), t \in [t_0, t_0 + T_e]$ are highly likely to be different from the predicted value $\boldsymbol{\xi}^*(t), t \in [t_0, t_0 + T_e]$. In the next step, the optimal control problem is solved again over a shifted horizon based on the new state measurements. The feedback of measurement information to the optimization endows the whole procedure with a robustness typical of closed-loop systems. This process is thus repeated at each step until terminal requirements are satisfied.

III. Nonlinear MPC Algorithm for Obstacle Avoidance

In this section, details of the nonlinear MPC algorithm for obstacle avoidance are presented. The nonlinear MPC algorithm consists of two parts: the LIDAR data processor and the control commands generator. Fig. 2 shows the block diagram with the nonlinear MPC algorithm and the AGV in the loop.

The LIDAR data processor simplifies the obstacle shape, adds safety margin, and partitions the safe region. The outputs of LIDAR data processor, the task information, and the estimated vehicle states are used in the formulation of the OCPs. The formulated OCPs are then solved and the control commands associated with the lowest cost solution is executed by the AGV. These steps are discussed in detail in the following sub-sections.

Three external inputs to the nonlinear MPC algorithm are required: task information, obstacle information, and estimated states. Within the scope of this paper, the task information is the specified target location and the desired vehicle speed,

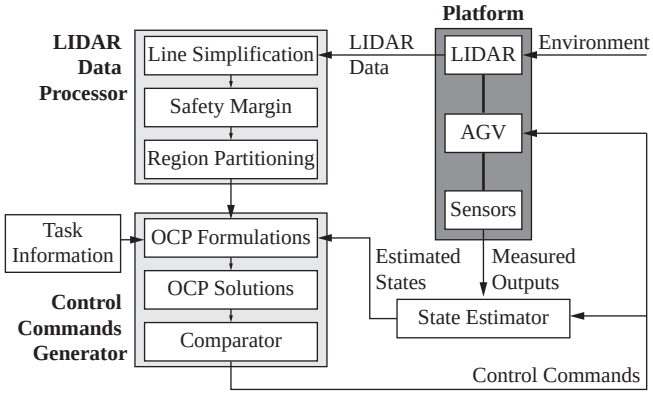
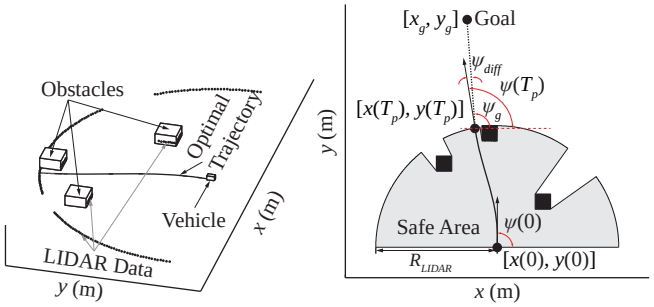


Fig. 2. Schematic of the nonlinear MPC algorithm



(a) 3D view (b) Top view

Fig. 3. (a) A sample obstacle field and the LIDAR detection data plotted in the 3D space; (b) the top view of the LIDAR data and the safe area

which is directly specified by the user. The task information could also be generated by a high-level global path planning algorithm.

The obstacle information is obtained from a LIDAR sensor, which provides information about range and geometrical characteristics of the closest objects to the vehicle. A 2D LIDAR sensor that is mounted in front of the vehicle is used. The LIDAR returns the distance to the closest obstacle boundary in each radial direction at an angular resolution of ϵ. The angular range is $[0°, 180°]$, with the vehicle heading direction being the $90°$ direction. For a direction without obstacles within the detection range, the LIDAR returns the maximum detection range R_{LIDAR} . Fig. 3 shows an obstacle field with three obstacles and the output of the LIDAR for the particular vehicle pose. It is assumed that all obstacles of interest are at least the height of where the LIDAR is mounted.

The vehicle states are required to properly initialize the vehicle model used in the algorithm. In a real application, a state estimator is needed to estimate the states, since not all states can be directly measured. However, in this paper, the AGV is simulated and the state estimator is ignored.

A. LIDAR Data Processor

In this section, the procedures included in the LIDAR data processor are described. The LIDAR data processor processes a sequence of points defining the safe region into specifications that can be used in the OCP formulation. In this paper, a 2D LIDAR is used and the set of points defining the safe region is simply the detected points by the LIDAR. If a 3D LIDAR or a camera is used to sense the environment, an algorithm is required to find the points that specify the safe region first.

1) Line Simplification: The first step of the data processing is to reduce the number of points that defines the obstacle boundaries for further processing. However, the points obtained directly from the LIDAR can be noisy because the obstacle boundaries are not smooth and the detection results are not exact. An algorithm is required to identify the minimum number of lines for approximating the sequence of points considering these noises. The Ramer-Douglas-Peucker algorithm is an algorithm for reducing the number of points in a curve that is approximated by a series of points, which is widely used to perform simplification and denoising of range data acquired by a LIDAR [21]. As illustrated in Fig. 4, the numerous detected points, which are generated using a simulated LIDAR with added noise, can be simplified into two line segments represented by three points, which are a good approximation to the boundaries of the obstacle.

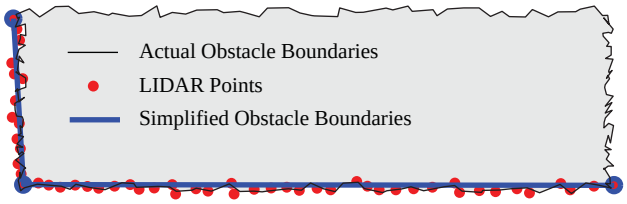


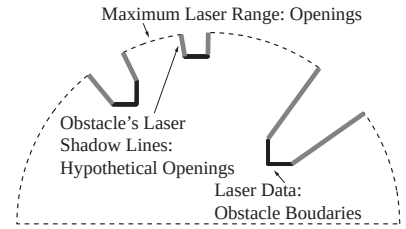
Fig. 4. An illustration of the line simplification algorithm.

2) Safety Margin: A safety margin, l_{SM}, is added to the safe region to account for the size of the vehicle, detection noises, and differences between the predicted trajectory and the actual trajectory. Adding a safety margin allows for ignoring the vehicle size in the OCP formulation. The safe area is a polygon in general and it is a simple polygon when the safe region boundary is from a 2D LIDAR sensor. Thus, algorithms for performing polygon offsetting (inflating/deflating) in computer graphics can be adopted. Specifically, the Vatti's clipping algorithm [22] implemented in the Clipper library developed by Angus Johnson [23] is used.

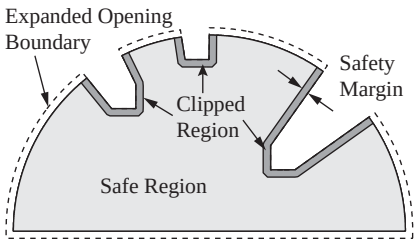
As shown in Fig. 5a, the boundary of the safe region consists of three types of segments:

- LIDAR data segments, which specify the boundaries of the obstacles.
- Maximum LIDAR detection range segments, which are directions free from obstacles and are called "openings".
- Obstacle's laser shadow lines, which are along the rays from the LIDAR and are called "hypothetical openings". The are called "hypothetical openings," because in its current position and orientation the vehicle cannot know whether it is an actual opening or not due to the obstacle blocking its view.

To add the safety margin, the openings are expanded by the amount of specified safety margin first. Then, the entire region is shrunk by the amount of specified safety margin. This is to add safety margin only on the obstacle boundaries and

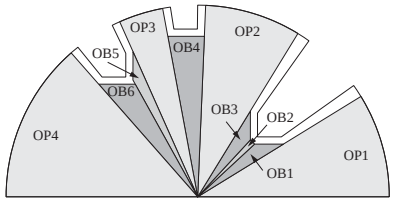


(a) Terminology

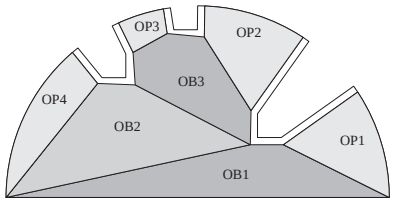


(b) Example

Fig. 5. (a) Three types of segments bounding the safe region; (b) an example of safe region with safety margin included



(a) Polar partitioning



(b) Convex partitioning

Fig. 6. Exemplified partitions using two approaches

hypothetical openings. Fig. 5b is an example of safe region with safety margin included.

3) Region Partitioning: The safe region exemplified in Fig. 5b is very difficult, if not impossible, to be defined using a single function. Even if the function exists, it is not differentiable at some points, which would cause problems in the OCP solver that requires all functions to be twice continuously differentiable. To address this challenge, the safe region is partitioned into several sub-regions and a multi-phase optimal control problem formulation is used. After partitioning, each sub-region can be specified by a set of inequalities. The functions involved in these inequalities are not piecewise functions and are differentiable.

There are many approaches for partitioning the safe region to meet the specified requirements. Two of them are introduced in this paper. One approach is named the "polar partitioning". The safe area is divided into sectors and triangles, where sectors are regions including an opening and triangles are regions including an obstacle boundary. Fig. 6a is the partitioning of the safe region shown in Fig. 5 using this approach. As an example, Region "OB4" is a triangle, which can be specified using three linear inequalities, whereas Region "OP3" is a sector, which is bounded by two lines. The third boundary of Region "OP3" is an arc; however, because of the limits on prediction horizon, this arc constraint will never be active and thus is ignored. This is an intuitive approach for partitioning the region from a 2D LIDAR sensor.

Another approach is named the "optimal convex partitioning" or simply "convex partitioning". The interior of this safe area is decomposed into a minimum number of convex regions without introducing additional points inside the polygon. Several algorithms exist in the literature for performing this task. The dynamic programming algorithm by Keil and Soneyink is incorporated [24], which is efficient in decomposing simple polygons. Fig. 6b is an exemplified partitioning. Similarly, all the regions can be specified using a set of linear inequalities after partitioning. This approach is capable of partitioning a safe region in a more general form.

In either approach, a sub-region can be defined by

$$\begin{bmatrix} \vdots & \vdots \\ a_j & b_j \\ \vdots & \vdots \end{bmatrix}^{(i)} \begin{bmatrix} x^{(i)} \\ y^{(i)} \end{bmatrix} \leq \begin{bmatrix} \vdots \\ c_j \\ \vdots \end{bmatrix}^{(i)} , j = 1, \ldots, L \qquad (1)$$

where i is the sub-region index and L is the total number of line segments bounding that sub-region; a_j, b_j and c_j are coefficients calculated based on the two end points of the corresponding line segments; (x, y) is a position in Cartesian coordinates.

Eq. (1) can be compacted in the following form:

$$\boldsymbol{A}^{(i)}_{L\times 1} x^{(i)}(t) + \boldsymbol{B}^{(i)}_{L\times 1} y^{(i)}(t) \leq \boldsymbol{C}^{(i)}_{L\times 1}, t \in [T^{i-1}, T^i] \qquad (2)$$

where $\boldsymbol{A}^{(i)}_{L\times 1}$ is a vector with the jth entry being a_j. The definitions of $\boldsymbol{B}^{(i)}_{L\times 1}$ and $\boldsymbol{C}^{(i)}_{L\times 1}$ are similar.

After partitioning, the entire safe region can be specified by a structure variable SafeRegion. The definition of the structure variable SafeRegion is given by the following pseudo-code.

```
int N; //number of sub-regions
int L[N]; //vector of number of line segments

struct SafeRegion {
   double AM[N][N]; //adjacency matrix
   SubRegion SR[N]; //subregion specifications
};

struct SubRegion {
   char Type; //type of the subregion: ''OP'', ''OB''
   int Index; //index of the subregion
   int HPNum; //number of hypothetical openings
   //end points of line segments (xs, ys, xe, ye)
   double EndPoints[L(Index)][4];
   //index of line segments representing
   //hypothetical openings
   double HPIndex[HPNum];
};
```

The adjacency matrix for the partitioning shown in Fig. 6b is given below. When two sub-regions have a common edge,

the corresponding entry in the matrix is set to 1; otherwise, it is set to 0.

$$\mathcal{AM}= \begin{array}{c} \\ \text{OP1} \\ \text{OP2} \\ \text{OP3} \\ \text{OP4} \\ \text{OB1} \\ \text{OB2} \\ \text{OB3} \end{array} \begin{array}{c} \begin{array}{ccccccc} \text{OP1} & \text{OP2} & \text{OP3} & \text{OP4} & \text{OB1} & \text{OB2} & \text{OB3} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

To avoid the obstacles and move towards the target, the trajectory should stay within the safe region and the last part of the predicted trajectory should lie within the sub-region of type "OP". The following list presents the procedures of using the structure variable SafeRegion in the OCP formulation to meet the above requirement. The safe region partitioning shown in Fig. 6b is used as an example to elaborate.

(a) Identify the first region to traverse ("SR"). Region "OB1" is the "SR" in this example, because the vehicle heading at the current position is pointing upwards.
(b) Identify all regions with a feasible opening ("TRs"). A feasible opening is an arc segment, which can have an intersection with a feasible trajectory over a slightly longer period than the prediction horizon. They can be identified using the two extreme trajectories, which are shown in Fig. 7. They are obtained by simulating the 2 DoF vehicle model using steering controls that are at the limits of handling at each step at the measured initial states. In the example, the arc segments in "OP2" and "OP3" are feasible openings and hence "OP2" and "OP3" are "TRs", whereas, "OP1" and "OP4" are not "TRs", because the vehicle cannot make a sharp enough turn to move into those partitions.
(c) Find the sequence of regions from the "SR" to a "TR" for all "TRs". This can be achieved by using Dijkstra's algorithm and the adjacency matrix [25]. For example, with "OP3" being the "TR", the region sequence is identified as "OB1 → OB2 → OB3 → OP3". The illustration of the region sequence is shown in Fig. 7. With "OP2" being the "TR", the region sequence is identified as "OB1 → OB2 → OB3 → OP2".

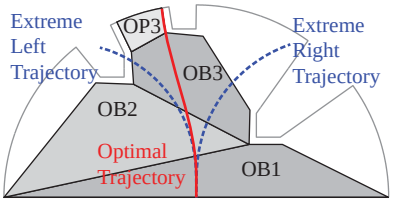


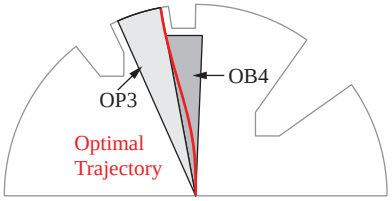
Fig. 7. An example of extreme trajectories and region sequence

For a region sequence from the "polar partitioning" as exemplified in Fig. 8a, a different region partition as shown in Fig. 8b can be obtained easily. This alternative partition approach is preferred when one of the boundaries separating two regions is almost along the vehicle heading direction.

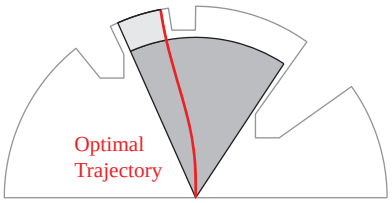
The specifications of the regions in this partition are given by

$$\begin{aligned} & R^{(i)}_{\min} \le \sqrt{[x^{(i)}(t)-x(0)]^2+[y^{(i)}(t)-y(0)]^2} \le R^{(i)}_{\max} \\ & \Phi^{(i)}_{\min} \le \operatorname{atan2}(y^{(i)}(t)-y(0), x^{(i)}(t)-x(0)) \le \Phi^{(i)}_{\max} \\ & t \in [T^{i-1}, T^i] \end{aligned} \tag{3}$$

where $R^{(i)}_{\min}$, $R^{(i)}_{\max}$, $\Phi^{(i)}_{\min}$, and $\Phi^{(i)}_{\max}$ are bounds calculated from the coordinates of end points specifying a region.



(a) Basic polar partitioning



(b) Modified polar partitioning

Fig. 8. Regions from polar partition approach and its variance

B. Control Command Generator

At each step of the MPC, a multi-phase optimal control problem is formulated for each of the "TRs". The number of phases is the number of regions from "SR" to the "TR". So, one or more OCPs is formulated and solved at each step. The formulation in general form is given by Eq. (4) - Eq. (10)

$$\underset{\boldsymbol{\xi},\boldsymbol{\zeta},T^1,\ldots,T^N}{\text{minimize}} \quad J = \mathcal{T}\left[\boldsymbol{\xi}^{(N)}\left(T^N\right), \boldsymbol{\zeta}^{(N)}\left(T^N\right), T^N\right] + \sum_{i=1}^{N}\left\{\int_{T^{i-1}}^{T^i} \mathcal{I}\left[\boldsymbol{\xi}^{(i)}(t), \boldsymbol{\zeta}^{(i)}(t)\right] \mathrm{d}t\right\} \tag{4}$$

$$\underset{\forall i=1,\ldots,N}{\text{subject to}} \quad \dot{\boldsymbol{\xi}}^{(i)}(t) = \boldsymbol{\mathcal{V}}\left(\boldsymbol{\xi}^{(i)}(t), \boldsymbol{\zeta}^{(i)}(t)\right) \tag{5}$$

$$\begin{aligned} & \boldsymbol{\xi}^{(i)}(T^{i-1}) = \boldsymbol{\xi}^{(i-1)}(T^{i-1}) \\ & \boldsymbol{\xi}^{(0)}(T^0) = \boldsymbol{\xi}_{\mathbf{0}} \end{aligned} \tag{6}$$

$$\boldsymbol{\mathcal{S}}^{(i)}(x^{(i)}(t), y^{(i)}(t)) \le 0 \tag{7}$$

$$\delta_{f,\min}(U_0) \le \delta_f^{(i)}(t) \le \delta_{f,\max}(U_0) \tag{8}$$

$$\varsigma_{f,\min} \le \varsigma_f^{(i)}(t) \le \varsigma_{f,\max} \tag{9}$$

$$\begin{aligned} & t \in [T^{i-1}, T^i], T^{i-1} < T^i \\ & T^0 = 0, T^N = T_p, T_{p,\min} < T_p \le T_{p,\max} \end{aligned} \tag{10}$$

By minimizing the cost function specified in Eq. (4), subject to constraints defined in Eq. (5) - Eq. (10) for all phases, the optimal state trajectories $\boldsymbol{\xi}^{(i)}(t), t \in [T^{i-1}, T^i]$, the optimal control trajectories $\boldsymbol{\zeta}^{(i)}(t), t \in [T^{i-1}, T^i]$, and the times T^{i-1}, T^i, $i = 1, \ldots, N$ are obtained, where N is the total number of phases.

Before we define the variables and explain the problem formulation in detail, let us go through the formulation at a high level. Eq. (5) is the dynamic model of the vehicle represented as a set of first order ordinary differential equations (ODEs). Eq. (6) sets the initial states of each phase as the final states of the previous phase. For the first phase, the initial states are the measured states. Eq. (7) defines the position constraints due to the obstacles perceived by the LIDAR sensor. A general form is given here. Eq. (8) and Eq. (9) represent the bounds on the steering angle and the steering rate, respectively. Eq. (10) specifies the time horizon for each phase. The overall prediction horizon is given by $t \in [0, T_p]$. There is a lower bound and an upper bound on the prediction time, which are described in Section IV.

1) Cost Function: The cost function defines the soft requirement, i.e., in what sense the trajectory is optimal. In this work, if the task is only to pass a target location without direction requirement, the trajectory is optimal when the end point of the predicted trajectory is close to the target, and the final heading angle is pointing to the target because a shorter distance-to-go is preferred. The cost function is defined as

$$J = \frac{s_T}{s_0} + w_\psi \psi_{\text{diff}}^2 + w_d d \tag{11}$$

where

$$s_0 = \sqrt{[x_g - x(0)]^2 + [y_g - y(0)]^2} \tag{12}$$

$$s_T = \sqrt{[x_g - x(T_p)]^2 + [y_g - y(T_p)]^2} \tag{13}$$

$$\psi_{\text{frg}} = \text{atan2}\left(y_g - y(T_p), x_g - x(T_p)\right) \tag{14}$$

$$\psi_{\text{diff}} = \text{atan2}\left(\sin(\psi(T_p) - \psi_{\text{frg}}), \cos(\psi(T_p) - \psi_{\text{frg}})\right) \tag{15}$$

$$d = \int_0^{T_p} \left[\varsigma_f^2(t) + w_\delta \delta_f^2(t)\right] \, \mathrm{d}t \tag{16}$$

Specifically, the cost function formulation includes three terms that are linearly combined using relative weights, w_ψ and w_d.

The first term is a ratio between distance s_T and distance s_0, where s_0 is the distance between the initial position $[x(0), y(0)]$ and the goal $[x_g, y_g]$ as defined in Eq. (12), and s_T is the distance between the end point of the predicted trajectory $[x(T_p), y(T_p)]$ and the goal as defined in Eq. (13). Visual representations of all variables are shown in Fig. 3b. The second term is the difference between the final heading angle $\psi(T_P)$ and the angle of the goal relative to the end point of the predicted trajectory ψ_{frg} as defined in Eq. (15). The third term is a regulation term minimizing the control effort d as defined in Eq. (16), where ς_f is the steering rate, which is the control command to be optimized, δ_f is the front wheel steering angle, and w_δ is a weight.

If a particular direction of passing the target location in global coordinates is also required, the following cost function is used

$$J = \frac{s_T}{s_0} + w_\psi \psi_{\text{diff}}^2 + w_s s + w_d d \tag{17}$$

where

$$s = \int_0^{T_p} \left[l_a x(t) + l_b y(t) + l_c\right]^2 \, \mathrm{d}t \tag{18}$$

$$\begin{aligned} l_a &= \sin(\psi_g) \\ l_b &= -\cos(\psi_g) \\ l_c &= -\sin(\psi_g) x_g + \cos(\psi_g) y_g \end{aligned} \tag{19}$$

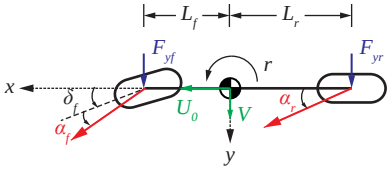
The cost function specified by Eq. (17) has one more term than Eq. (11). This term is to minimize the integral of the distance to the line given by $l_a x + l_b y + l_c = 0$ over the prediction horizon. This line is passing through the goal $[x_g, y_g]$ along the desired direction ψ_g.

When the target position is within the sensor's detection range, the term $\frac{s_T}{s_0}$ and $w_\psi \psi_{\text{diff}}^2$ are removed from the cost functions given in Eq. (11) and Eq. (17). Instead the following constraints are added to the OCP formulation.

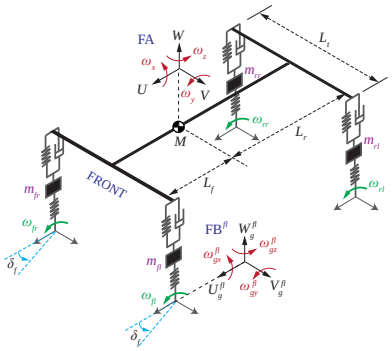
$$\begin{aligned} x_g - \sigma \le x(T_p) \le x_g + \sigma \\ y_g - \sigma \le y(T_p) \le y_g + \sigma \end{aligned} \tag{20}$$

where σ is a small margin. If the vehicle is within this margin from the target position, then the target is considered to be reached.

2) Vehicle Models: Two different vehicle dynamics models are used in this work: a 14 DoF model to represent the plant and to generate offline the dynamic-safety-related look-up tables used in the MPC, and a 2 DoF model used in the MPC to predict trajectories. The schematics of the two representations are given in Fig. 9. The 14 DoF model consists of a single sprung mass connected to four unsprung masses. The suspensions between the sprung mass and unsprung masses are modeled as spring-damper systems. In the 2 DoF model, the left and right tires on each axle are lumped together. The equations for the 14 DoF model are omitted here for space limitations, but can be found in the literature [26].



(a) Two DoF vehicle model



(b) Fourteen DoF vehicle model

Fig. 9. Schematic of the vehicle models

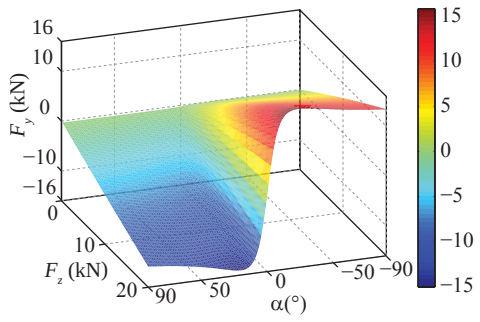


Fig. 10. Lateral tire force described by the Pacejka tire model.

The 2 DoF model is described by the following ODEs

$$\dot{V} = (F_{y,f} + F_{y,r})/M - U_0 r \tag{21}$$
$$\dot{r} = (F_{y,f} L_f - F_{y,r} L_r)/I_{zz} \tag{22}$$
$$\dot{\psi} = r \tag{23}$$
$$\dot{x} = U_0 \cos\psi - (V + L_f r)\sin\psi \tag{24}$$
$$\dot{y} = U_0 \sin\psi + (V + L_f r)\cos\psi \tag{25}$$

where $F_{y,f}$ and $F_{y,r}$ are tire lateral forces generated at the front axle and the rear axle, respectively. U_0 and V are the longitudinal speed and lateral speed in the vehicle's coordinate frame, respectively. r is the yaw rate, ψ is the yaw angle, (x, y) is the vehicle's front center location in global coordinates, M is the vehicle mass, I_{zz} is the moment of inertia, L_f is the distance between the front axle and the vehicle's CoG location, and L_r is the distance between the rear axle and the vehicle's CoG location.

By using the Pacejka Magic Formula ($\mathcal{MF}$) tire model with pure slip, the lateral tire forces are presented as

$$F_{y,f} = \mathcal{MF}_y(F_{z,f}, \alpha_f) \tag{26}$$
$$F_{y,r} = \mathcal{MF}_y(F_{z,r}, \alpha_r) \tag{27}$$

The exact form of the Pacejka tire model can be found in [27]. Fig. 10 shows the relationship between the tire lateral forces and the slip angle at different vertical loads described by the Pacejka tire model. In the 14 DoF vehicle model, Pacejka Magic Formula tire model with combined slip is used.

As concluded in [20], it is important to account for the longitudinal load transfer in the 2 DoF vehicle model when the vehicle travels at high speed. Thus, the following relationships are used with the 2DoF model to calculate the vertical loads on the front and rear axles taking into account the longitudinal load transfer effects:

$$F_{z,f} = \frac{MgL_r + MVrh_{\mathrm{CG}}}{L_f + L_r} \tag{28}$$
$$F_{z,r} = \frac{MgL_f - MVrh_{\mathrm{CG}}}{L_f + L_r} \tag{29}$$

where h_{CG} is the height of the vehicle CoG location above the ground.

In addition, the slip angles of front and rear tires are obtained from

$$\alpha_f = \tan^{-1}\left(\frac{V + L_f r}{U_0} - \delta_f\right) \tag{30}$$
$$\alpha_r = \tan^{-1}\left(\frac{V - L_r r}{U_0}\right) \tag{31}$$

The steering rate ς_f is used as the control command to be optimized and the steering angle δ_f is set as an additional state variable of the system. The reason is to make it possible to obtain a smooth steering angle sequence and impose a limit on the steering rate.

By setting the state vector as $\boldsymbol{\xi} = [\ x \quad y \quad \psi \quad V \quad r \quad \delta_f\ ]$ and the control vector as $\boldsymbol{\zeta} = \varsigma_f$, the state-space equation for the 2 DoF nonlinear vehicle model can be written as

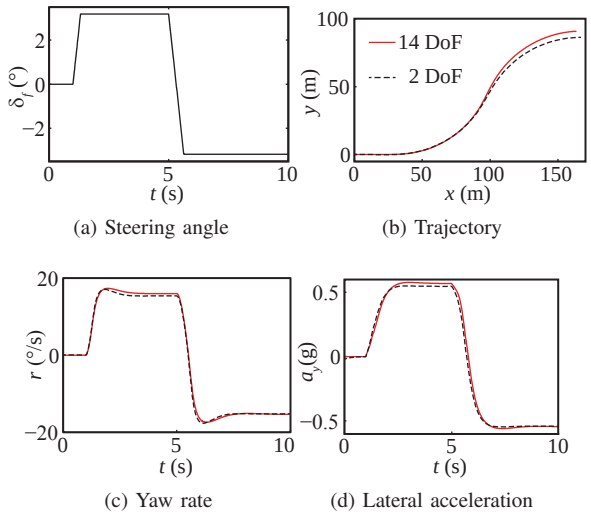
$$\dot{\boldsymbol{\xi}} = \boldsymbol{f}(\boldsymbol{\xi}) + \boldsymbol{B}\boldsymbol{\zeta} \tag{32}$$

where

$$\boldsymbol{f}(\boldsymbol{\xi}) = \begin{bmatrix} U_0\cos\psi - (V + L_f r)\sin\psi \\ U_0\sin\psi + (V + L_f r)\cos\psi \\ r \\ (F_{y,f} + F_{y,r})/M - U_0 r \\ (F_{y,f}L_f - F_{y,r}L_r)/I_{zz} \\ 0 \end{bmatrix}$$

$$\boldsymbol{B}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The steering profile in Fig. 11a is used as the input and a vehicle speed of 20 m/s is considered for a model comparison. As shown by the comparison of trajectories in Fig. 11b, yaw rates in Fig. 11c, and lateral accelerations in Fig. 11d, the 2 DoF vehicle model with longitudinal load transfer and nonlinear tire model is a very good approximation to the 14 DoF vehicle model.



(a) Steering angle (b) Trajectory

(c) Yaw rate (d) Lateral acceleration

Fig. 11. Comparison between the 2 DoF vehicle model and the 14 DoF vehicle model

3) Obstacle Avoidance: Obstacle avoidance is enforced through the constraint that the vehicle trajectory must lie within the safe region. For each of the phases in the multiphase OCP, the vectors $\boldsymbol{A}^{(i)}_{L\times 1}$, $\boldsymbol{B}^{(i)}_{L\times 1}$, and $\boldsymbol{C}^{(i)}_{L\times 1}$ or the bounds $R^{(i)}_{\min}$, $R^{(i)}_{\max}$, $\Phi^{(i)}_{\min}$, and $\Phi^{(i)}_{\max}$ can be calculated using the values stored

in the structure variable SafeRegion. The specific form of Eq. (7) is either given by Eq. (2) or Eq. (3) depending on the scenario.

Regarding the safe region partitioning, there are still some questions, such as, what are the characteristics of a good partition, how to evaluate the goodness of the partition, and how to generate a good partition systematically and efficiently. Addressing these questions requires future work. In the simulations for this paper, the convex partition approach is used primarily, and the polar partition approach is used secondarily, if needed.

4) Dynamical Safety: In this study, ensuring the vehicle's dynamical safety is translated to avoiding single tire lift-off. This is a conservative criterion used to prevent rollover [28]. This requirement could be taken into account directly to enforce a positive vertical load on all four tires at all times. However, vehicle models that could predict the vertical tire loads on all four tires would require a level of complexity whose computational load would be prohibitively high for the purpose of MPC. Therefore, another conservative approximation of the dynamical safety requirement is considered; namely, an upper bound on the steering angle magnitude as expressed by the following inequality constraint

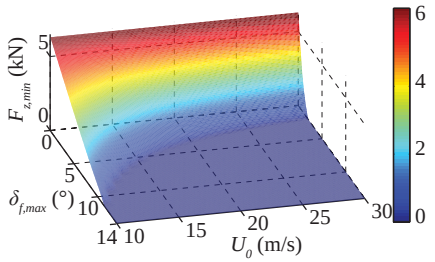
$$|\delta_f(t)| \leq \delta_{f,\max}(U_0) \tag{33}$$

where the maximum steering angle $\delta_{f,\max}$ is a function of only the vehicle speed when the vehicle is assumed to move on a flat surface.

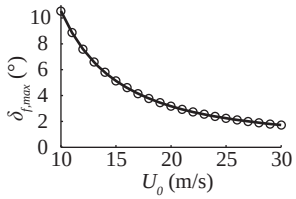
For all combinations of longitudinal speed ranging from 10 m/s to 30 m/s and maximum steering angle ranging from 0° to 14°, the corresponding minimum tire vertical loads are obtained using the 14 DoF vehicle model. The relationship is shown in Fig. 12a. If a minimum vertical load threshold is set, the relationship between the maximum steering angle and longitudinal speed can be extracted. Fig. 12b shows the relationship when $F_{z,\min}$ is set as 500 N.

5) Solution Techniques: The formulated nonlinear multi-phase optimal control problems are solved using a two-step procedure. First, the continuous-time OCP is transcribed into to a nonlinear programming (NLP) problem using a direct method called *hp*-pseudospectral method [29], [30], [31]. Second, the resulting NLP problem is solved using the interior point method [32].

The *hp*-pseudospectral method discretizes a continuous-time OCP into an NLP problem by approximating the state and control using a variable number of approximating intervals and variable-degree polynomial approximations of them within each interval. The differential-algebraic constraints of the OCP are enforced at a finite set of collocation points, where the collocation points are Legendre-Gauss-Radau (LGR) quadrature points. This method has been shown to be able to accurately approximate the solution to a general continuous-time OCP in a computationally efficient manner [31].



(a)



(b)

Fig. 12. (a) Minimum tire vertical load at different combinations of vehicle speed and maximum steering angle; (b) maximum steering angle as a function of vehicle longitudinal speed when the minimum vertical load threshold is 500 N.

After transforming from the time interval $t \in [0, T_p]$ to the time interval $\tau \in [-1, 1]$ via the following variable transformation

$$t = \frac{T_p}{2}\tau + \frac{T_p}{2} \tag{34}$$

the state $\boldsymbol{\xi}$ is approximated by a polynomial of degree at most n as follows

$$\boldsymbol{\xi}(\tau) \approx \sum_{i=1}^{n+1} \boldsymbol{\xi}_i L_i(\tau), L_i(\tau) = \prod_{j=1, j\neq i}^{n+1} \frac{\tau - \tau_j}{\tau_i - \tau_j} \tag{35}$$

where $\tau_i (i = 1, \ldots, n)$ is the LGR collocation points, $L_i(\tau)(i = 1, \ldots, n)$ is a basis of Lagrange polynomials, and $\boldsymbol{\xi}_i$ is the state approximation at τ_i.

To solve the NLP problem, a primal-dual interior-point algorithm with a filter line search method implemented in IPOPT is used [32]. The basic idea of the interior point method is to decompose the NLP problem with both equality and inequality constraints into a sequence of equality constrained problems by introducing a barrier function and barrier parameter. The NLP problem with only equality constraints can then be solved iteratively. The search direction is determined using the Newton-Raphson method and the step size is obtained using the backtracking line search.

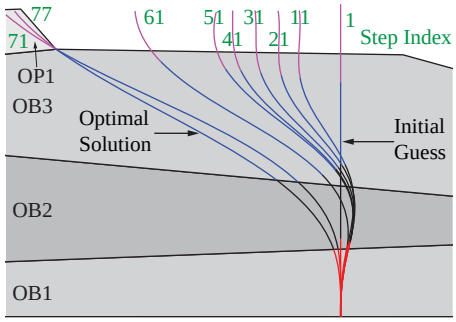
The interior point method converts the general NLP problem given by Eq. (36) to a series of NLPs with only equality constraints given by Eq. (37).

$$\begin{aligned} &\underset{Z\in\mathbb{N}^n}{\text{minimize}} && f(Z) \\ &\text{subject to} && \boldsymbol{C}(Z) = 0 \\ & && Z \geq 0 \end{aligned} \tag{36}$$

$$\begin{aligned} &\underset{Z\in\mathbb{N}^n}{\text{minimize}} && f(Z) + \mu_k B(Z) \\ &\text{subject to} && \boldsymbol{C}(Z) = 0 \end{aligned} \tag{37}$$

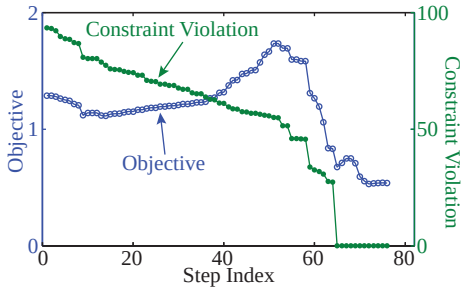
where μ is a small positive scalar called "barrier parameter". As μ converges to zero, the solution to Eq. (37) should converge to a solution to Eq. (36). $B(.)$ is a barrier function.

As an example, Fig. 13a shows the trajectory iterations in solving the problem given in Fig. 7. This is a four-phase problem. The initial guess is a straight line assuming equal length at each phase, which is not a feasible solution. Nevertheless, after 77 iterations, the solution converges to the optimal solution. Fig. 13b shows the corresponding objective value and maximum constraint violation at all steps.



Note: figure stretched along the horizontal direction

(a) Solution iterations



(b) Objective and constraint violation

Fig. 13. (a) The trajectory iterations from the initial guess to the optimal solution; (b) the objective and constraint violation at all steps

IV. Simulation Results and Discussion

In this section, numerical simulations of the developed nonlinear MPC obstacle avoidance algorithm with a 14 DoF vehicle model as the plant are presented. Table I gives part of the vehicle parameters used by the 14 DoF vehicle model and all the parameters used by the 2 DoF vehicle model.

TABLE I
Vehicle Parameters

Parameter	Value
M (kg)	2252
m (kg)	110
I_{zz} (kg-m^2)	4110
L_f (m)	1.58
L_r (m)	1.72
L_t (m)	1.82
$h_{\text{CG}}(m)$	1.00

Three scenarios are considered in this section. In the first scenario, the vehicle is required to move from its initial location to a target location with the final heading angle required to be the same as its initial heading angle. Two obstacles are between the two locations. Vehicle speed ranging from 10 m/s to 30 m/s are considered.

In the second scenario, the vehicle has to traverse a dense obstacle field to reach the target location. There are 50 obstacles and each of them is 10 m $\times$ 10 m in size. The vehicle longitudinal speed is maintained at 20 m/s and there is no constraint on the final heading angle.

In the third scenario, the vehicle performs a double lane change maneuver at 15 m/s using the obstacle avoidance algorithm.

A. Scenario 1: Various Speeds

Table II summarizes the parameters of the nonlinear MPC algorithm, including weights in the cost function, safety margin, LIDAR detection range, length of prediction horizon, length of execution horizon, and maximum steering angle. The cost function given by Eq. 17 is used because the angle of passing the goal is specified. In the settings, the following relationship, which is used to ensure that all the predicted trajectories lie within the LIDAR detection range, is used

$$T_{p,\max} = \frac{R_{\text{LIDAR}}}{U_0} \tag{38}$$

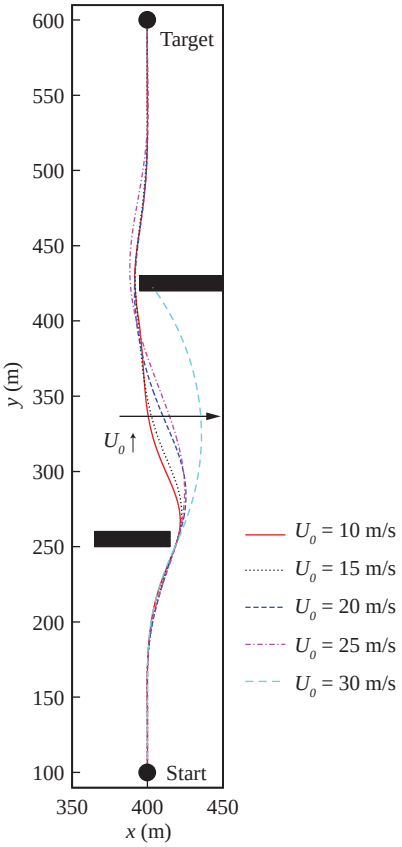
TABLE II
Simulation Parameters

U_0 (m/s)	10	15	20	25	30	30
w_ϕ (-)			1			1
w_d (-)			10			10
w_δ (-)			0.1			0.1
w_s (-)			10^{-4}			10^{-4}
l_{SM} (m)			3			3
R_{LIDAR} (m)			100			140
$T_{p,\max}$ (s)	10.0	6.7	5.0	4.0	3.3	4.7
T_e (s)	0.67	0.44	0.33	0.27	0.22	0.31
$\varsigma_{f,\max}$ ($^\circ/s$)			10			10
$\delta_{f,\max}$ ($^\circ$)	10.5	5.14	3.18	2.24	1.72	1.72

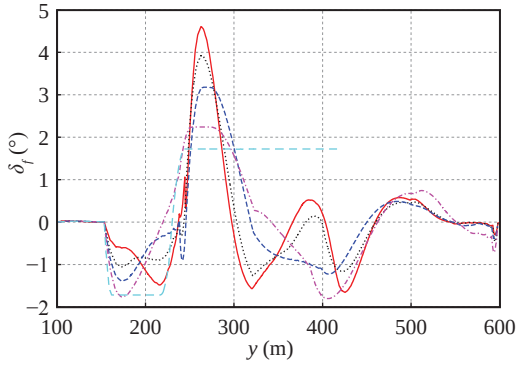
The first set of simulations uses a LIDAR with detection range of 100 m. The results of the simulations are presented in Fig. 14.

These results show that the developed algorithm can successfully navigate the vehicle through the specified obstacle field at 10 m/s, 15 m/s, 20 m/s, and 25 m/s. At these speeds, the vehicle avoids all obstacles, passes the target from the desired direction, and is dynamically safe as shown in Fig. 14c. However, the vehicle hits the second obstacle when the longitudinal speed is maintained at 30 m/s. This is because at this speed, the vehicle is not capable of making a turn at a smaller radius safely. A threshold of 500 N is set on the minimum tire vertical load and the corresponding maximum steering angle is set as a hard constraint in the OCP formulation. This constraint is active at most of the time during the maneuver as shown in Fig. 14b.

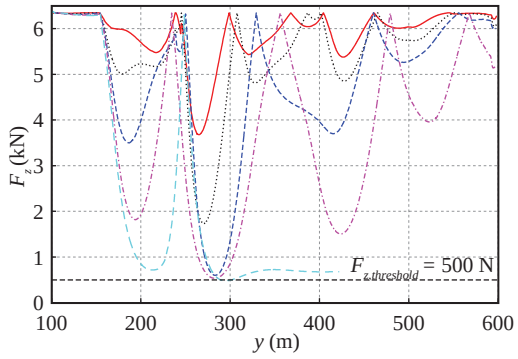
The navigation at 30 m/s fails because the LIDAR detection range is not long enough and hence the prediction horizon



(a) Trajectory

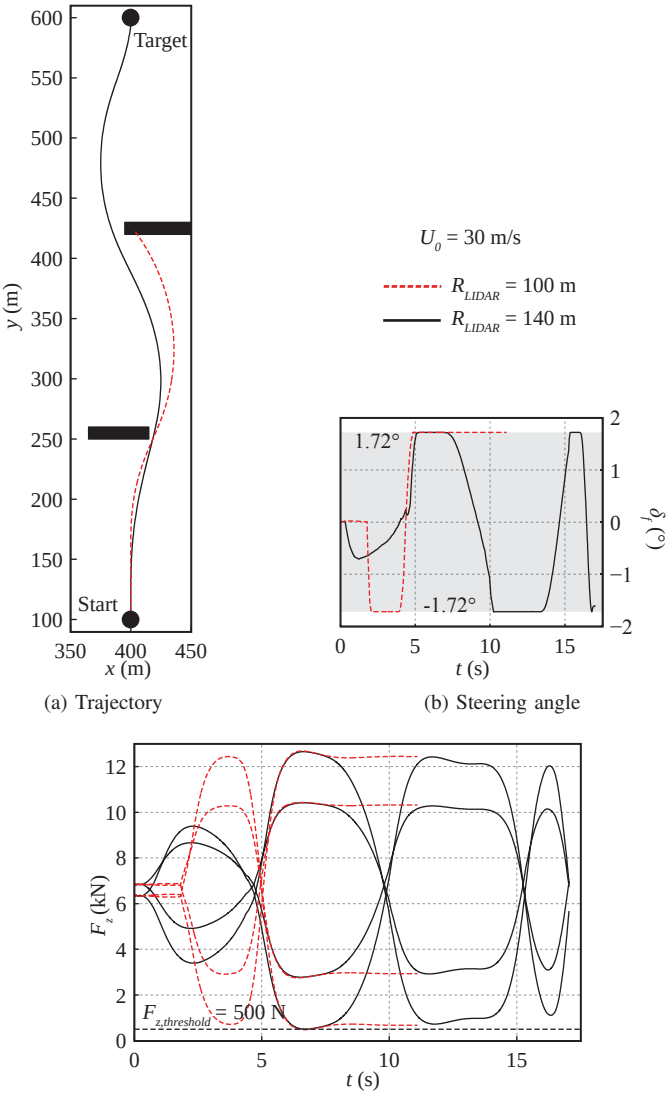


(b) Steering angle



(c) Minimum tire vertical load

Fig. 14. Results of simulations with various longitudinal speed



(a) Trajectory

(b) Steering angle

(c) Tire vertical loads

Fig. 15. Results of simulations with different LIDAR detection ranges at 30 m/s

is too short to prepare the vehicle to avoid the obstacles sufficiently early. Fig. 15 shows the results of simulations with different LIDAR detection ranges. When a longer detection range of 140 m and a longer prediction horizon are used, the vehicle travels through the field safely.

Because the on-board sensors provide information about the environment only within the close proximity of the vehicle, the obstacle avoidance algorithm is not capable of determining the maximum speed that can be used to safely navigate through an obstacle field. The maximum speed should come from a high-level planner. Alternatively, a conservative lower bound on the prediction horizon or a conservative lower bound on the sensor detection range can be imposed to ensure that the obstacle avoidance maneuver is performed early enough.

These limits could be obtained from the trajectory for making a 90° turn when the initial steering angle is at the minimum bound, which is considered as the most extreme maneuver. The trajectories from speeds ranging from 10 m/s to 30 m/s are shown in Fig. 16. The time of completing

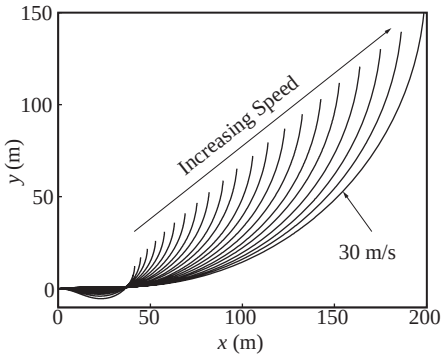
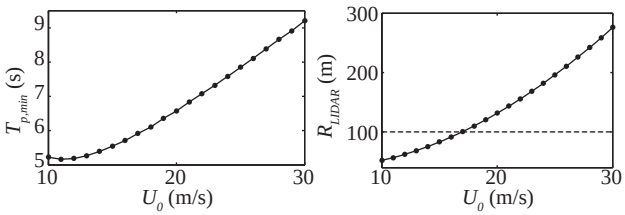


Fig. 16. The trajectories of vehicle making a 90° turn at various speeds



(a) Minimum prediction time (b) Minimum detection range

Fig. 17. Limits on parameters to ensure early maneuver

this maneuver is considered as the minimum prediction time, which is summarized in Fig. 17a. According to Eq. (38), the minimum detection range is given by Fig. 17b. As shown in the figure, the speed should be limited below 17 m/s when the LIDAR detection range is 100 m if this conservative bound is used.

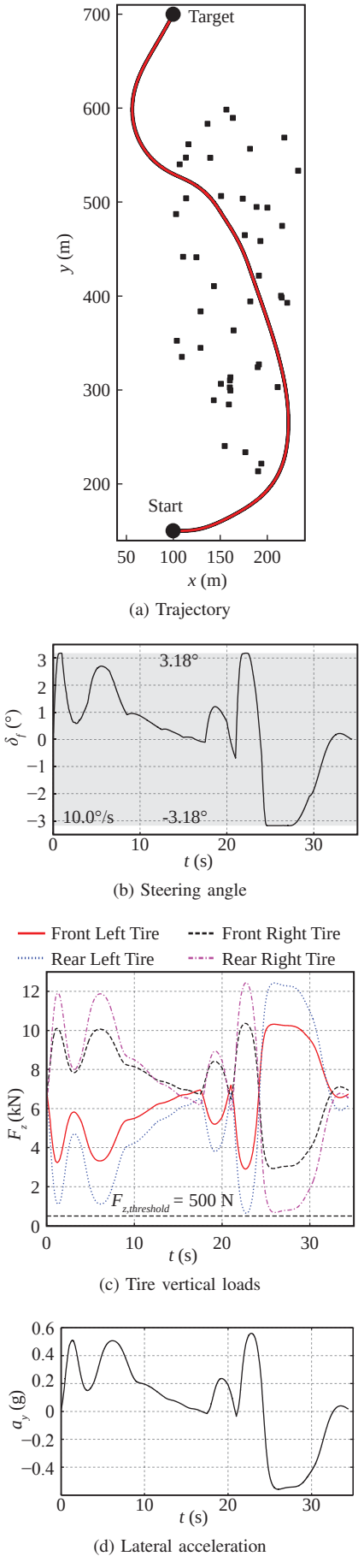
B. Scenario 2: Dense Obstacle Field

This simulation is to test the capability of the algorithm within a dense obstacle field. In this simulation, the vehicle speed is maintained at 20 m/s and there is no constraint on the final heading angle. Hence, Eq. (11) is used as the cost function and the simulation parameters as the same as the one corresponding to 20 m/s in Table II except that w_s is not used. Fig. 18 shows the simulation results. The vehicle clears the obstacle field and reaches the target successfully using the algorithm.

In this scenario, at most of the steps, there are multiple feasible openings as exemplified by Fig. 19. For each of the feasible openings, an OCP is formulated and solved. After all of them are solved, their objective function values are compared and the one with the smallest value is considered the best solution. In this example, the objective values of the calculated trajectories from right to left are 0.76, 0.74, 0.92, respectively. The smallest one is 0.74 and the control commands corresponding to the trajectory in the middle is sent to the plant.

C. Scenario 3: Double Lane Change

The last simulation is to test the capability of the algorithm in an on-road scenario. The vehicle performs a double lane change maneuver at 15 m/s using the nonlinear MPC



(a) Trajectory

(b) Steering angle

(c) Tire vertical loads

(d) Lateral acceleration

Fig. 18. Simulation results of navigation within dense obstacle field

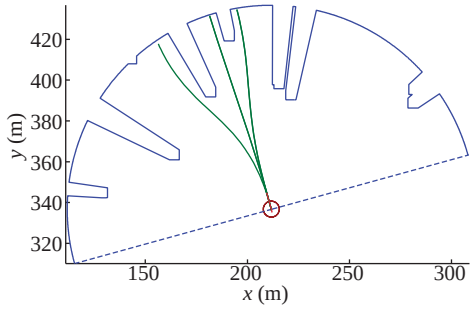


Fig. 19. Multiple feasible openings.

TABLE III
SIMULATION PARAMETERS FOR DOUBLE LANE CHANGE TEST

Parameter	Value
w_ϕ (-)	10
w_d (-)	10
w_δ (-)	0.1
w_s (-)	10^{-2}
l_{SM} (m)	1.6
R_{LIDAR} (m)	50
$T_{p,\max}$ (s)	3.0
T_e (s)	0.3
$\varsigma_{f,\max}$ (°/s)	10
$\delta_{f,\max}$ (°)	5.14

algorithm. Table III summarizes the parameters used. In this simulation, the weights w_d and w_s are increased to increase the vehicle's tendency of following the original lane.

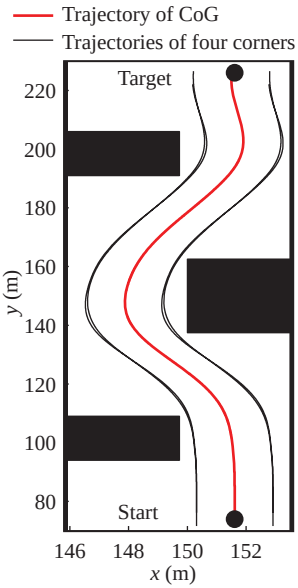
Fig. 20 shows the generated trajectory and the corresponding steering angle. Fig. 20a shows the trajectory of the CoG of the plant and the corresponding trajectories of the four corners of the vehicle. It can be seen that all the trajectories are within the white space, which means that the vehicle is free from collision. Fig. 20b is the corresponding steering sequence.

In this scenario, in most of the steps, there are no "openings" as defined in Fig. 5a. But there are "hypothetical openings", which are lines connecting an obstacle and an opening that are long enough. The "TRs" are then defined as all regions with a feasible hypothetical opening. Fig. 21 shows the use of a hypothetical opening.

V. CONCLUSION

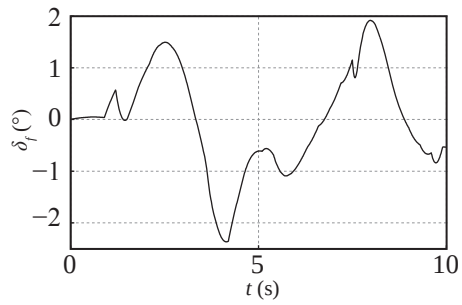
This paper presents the development of a novel nonlinear MPC algorithm for obstacle avoidance in autonomous ground vehicles using real-time sensor for environment detection. A multi-phase optimal control problem formulation is used to incorporate data from the on-board LIDAR sensor and the dynamic limitations of the vehicle to find an optimal solution. The resulting problem is solved using the pseudo-spectral method and the interior point method. Simulation results show that the developed method can yield a satisfactory performance under various scenarios.

The limitations of this work can be summarized as follows. The vehicle is assumed to travel on a flat terrain at a constant



Aspect Ratio = 1:10

(a) Trajectory



(b) Steering angle

Fig. 20. Simulation results of double lane change maneuver

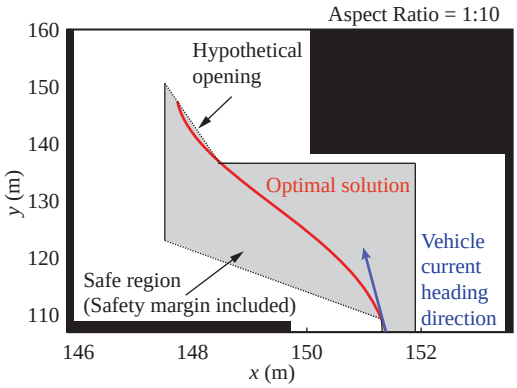


Fig. 21. The usage of hypothetical opening

speed. It is of interest to consider 3D terrains and include the vehicle speed as a second controlled variable. Furthermore, uncertainties in the model or sensor measurements are not yet considered, nor are moving obstacles. Addressing these questions is subject to future work. In addition, further investigations are required to study the impact of the objective function on the algorithm's performance systematically and develop an approach for adapting the weights in the objective function based on sensor measurements.

REFERENCES

[1] C. E. Beal and J. C. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling", *IEEE Transactions on Control Systems Technology*, vol. 21, pp. 1258-1269, 2013.

[2] S. M. LaValle, *Planning algorithms*, Cambridge University Press, 2006.

[3] B. D. Luders, S. Karaman, and J. P. How, "Robust sampling-based motion planning with asymptotic optimality guarantees," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference (GNC)*, 2013.

[4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, pp. 90-98, 1986.

[5] S. Shimoda, Y. Kuroda, and K. Iagnemma, "High-speed navigation of unmanned ground vehicles on uneven terrain using potential fields," *Robotica*, vol. 25, pp. 409-424, 2007.

[6] A. Hussein, H. Mostafa, M. Badrel-Din, O. Sultan, and A. Khamis, "Meta-heuristic optimization approach to mobile robot path planning," in *Proceedings of the International Conference on Engineering and Technology (ICET)*, 2012.

[7] P. Ogren and N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," *IEEE Transactions on Robotics*, vol. 21, pp. 188-195, 2005.

[8] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *Proceedings of the Dynamic Systems and Control Conference (DSCC)*, 2010.

[9] F. Allgwer and A. Zheng, *Nonlinear model predictive control* vol. 26: Springer, 2000.

[10] J. M. Park, D. W. Kim, Y. S. Yoon, H. J. Kim, and K. S. Yi, "Obstacle avoidance of autonomous vehicles based on model predictive control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 223, pp. 1499-1516, 2009.

[11] G. P. Bevan, H. Gollee, and J. O'Reilly, "Trajectory generation for road vehicle obstacle avoidance using convex optimization," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 224, pp. 455-473, 2010.

[12] A. Tahirovic and G. Magnani, "General framework for mobile robot navigation using passivity-based MPC," *IEEE Transactions on Automatic Control*, vol. 56, pp. 184-190, 2011.

[13] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, H. E. Tseng, and F. Borrelli, "Predictive control for agile semi-autonomous ground vehicles using motion primitives," in *Proceedings of the American Control Conference (ACC)*, 2012, pp. 4239-4244.

[14] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl., "An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles," in *Proceedings of the European Control Conference (ECC)*, 2013.

[15] J. H. Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *Proceedings of the American Control Conference (ACC)*, 2013, pp. 188-193.

[16] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "A model predictive control approach for combined braking and steering in autonomous vehicles," in *Proceedings of the Mediterranean Conference on Control and Automation (MED)*, 2007, pp. 1-6.

[17] S. J. Anderson, S. C. Peters, T. E. Pilutti, and K. Iagnemma, "An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios," *International Journal of Vehicle Autonomous Systems*, vol. 8, pp. 190-216, 2010.

[18] A. Gray, M. Ali, Y. Gao, J. K. Hedrick, and F. Borrelli, "A unified approach to threat assessment and control for automotive active safety," *IEEE Transactions On Intelligent Transportation Systems*, vol. 14, pp. 1490-1499, 2013.

[19] J. Liu, P. Jayakumar, J. L. Overholt, J. L. Stein, and T. Ersal, "The role of model fidelity in model predictive control based hazard avoidance in unmanned ground vehicles using LIDAR sensors," in *Dynamic Systems and Control Conference (DSCC)*, 2013.

[20] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "A multi-stage optimization formulation for mpc-based obstacle avoidance in autonomous vehicles using a LIDAR sensor," in *Dynamic Systems and Control Conference (DSCC)*, 2014.

[21] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, pp. 112-122, 1973.

[22] B. R. Vatti, "A generic solution to polygon clipping," *Communications of the ACM*, vol. 35, pp. 56-63, 1992.

[23] A. Johnson, "Clipper - an open source freeware library for clipping and offsetting lines and polygons". Retrieved September, 2014 Available: http://www.angusj.com/delphi/clipper.php

[24] M. Keil and J. Snoeyink, "On the time bound for convex decomposition of simple polygons," *International Journal of Computational Geometry and Applications*, vol. 12, pp. 181-192, 2002.

[25] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, pp. 269-271, 1959.

[26] T. Shim and C. Ghike, "Understanding the limitations of different vehicle models for roll dynamics studies," *Vehicle System Dynamics*, vol. 45, pp. 191-216, 2007.

[27] H. B. Pacejka, *Tire and vehicle dynamics*, Elsevier, 2005.

[28] B.-C. Chen and H. Peng, "Rollover warning for articulated heavy vehicles based on a time-to-rollover metric," *Journal of Dynamic Systems, Measurement and Control*, vol. 127, pp. 406-414, 2005.

[29] A. V. Rao, "A survey of numerical methods for optimal control," in *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, 2010, pp. 497-528.

[30] C. L. Darby, W. W. Hager, and A. V. Rao, "An hp-adaptive pseudospectral method for solving optimal control problems," *Optimal Control Applications and Methods*, vol. 32, pp. 476-502, 2011.

[31] C. L. Darby, "hppseudospectral method for solving continuous-time nonlinear optimal control problems," *Ph.D. Thesis*, University of Florida, 2011.

[32] A. Waechter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25-57, 2006.

Jiechao Liu received the B.S.E. degree in electrical and computer engineering from Shanghai Jiao Tong University, China, in 2009, and the B.S.E. and M.S.E. degrees in mechanical engineering from the University of Michigan, Ann Arbor, MI, in 2009 and 2011, respectively. He is currently a Ph.D. candidate in the Department of Mechanical Engineering, University of Michigan, Ann Arbor. He is currently with the Automated Modeling Laboratory, University of Michigan, Ann Arbor, where he is researching methods in vehicle-dynamics-conscious real-time hazard avoidance in autonomous ground vehicles. His main research discipline is in system dynamics and control, and his research interests include vehicle dynamics and optimal control.

Paramsothy Jayakumar is a Senior Research Scientist, SAE Fellow, and a member of the Analytics Team at the U.S. Army Tank Automotive Research, Development, & Engineering Center (TARDEC) in Warren, Michigan. Prior to joining U.S. Army TARDEC, he worked for BAE Systems, Ford Motor Company, Altair Engineering, and Engineering Mechanics Research Corporation in the areas of multibody dynamics software development, vehicle dynamics modeling & simulation consulting, simulation technology development, durability load simulation, vehicle instrumentation & loads measurement, and road load engineering. Jayakumar has written more than 100 technical publications including journal articles and conference papers. His research in terramechanics and multibody dynamics won the best paper awards at the National Defense Industrial Association's Ground Vehicle Systems Engineering and Technology Symposium in 2011 and 2012. He holds a U.S. patent for a system for virtual prediction of road loads and tire modeling. He was also instrumental in developing seven SAE standards for tire testing for the purpose of tire modeling for which he received the SAE 2014 James M. Crawford Technical Standards Board Outstanding Achievement Award. Jayakumar is a member of the U.S. Army Acquisition Corps, an Honorary Fellow of the Department of Mechanical Engineering at the University of Wisconsin Madison, and an Associate Editor for the ASME Journal of Computational and Nonlinear Dynamics. He received his M.S. and Ph.D. degrees in structural dynamics from Caltech, and B.Sc. Eng. (Hons, First Class) from the University of Peradeniya, Sri Lanka.

Jeffrey L. Stein received the B.S. degree in premedical studies from the University of Massachusetts, Amherst, MA, in 1973, and the S.B., S.M, and Ph.D. degrees in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, MA, in 1976, 1976, and 1983, respectively. Since 1983 he has been with the University of Michigan, Ann Arbor, MI, where he is currently a Professor of Mechanical Engineering. His research interests include computer based modeling and simulation tools for system design and control, with applications to vehicle-to-grid integration, vehicle electrification, conventional vehicles, machine tools, and lower leg prosthetics. He has particular interest in algorithms for automating the development of proper dynamic mathematical models, i.e., minimum yet sufficient complexity models with physical parameters.

Tulga Ersal received the B.S.E. degree from the Istanbul Technical University, Istanbul, Turkey, in 2001, and the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI USA, in 2003 and 2007, respectively, all in mechanical engineering. He is currently an Assistant Research Scientist in the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI. His research interests include modeling, simulation, and control of dynamic systems, with applications to energy systems, multibody dynamics, vehicle systems, and biomechanics. Dr. Ersal is a member of the ASME. He is the recipient of Deans and Presidents Awards at the Istanbul Technical University, the Siemens Excellence Award, and three merit fellowships at the Istanbul Technical University and the University of Michigan. He also received the semi-plenary paper award at the 2012 Dynamic Systems and Control Conference.